DOI:
10.1140/epjds/s13688-018-0159-3

Document Version
Publisher's PDF, also known as Version of record

Pappalardo et al. *EPJ Data Science* (2018) 7:30
https://doi.org/10.1140/epjds/s13688-018-0159-3

EPJ Data Science
a SpringerOpen Journal

REGULAR ARTICLE Open Access

Blockchain inefficiency in the Bitcoin peers network

Giuseppe Pappalardo[1*], Tiziana Di Matteo[2,3,4], Guido Caldarelli[1,5,6,7] and Tomaso Aste[2,3]

*Correspondence:
giuseppe.pappalardo@imtlucca.it
[1]IMT Alti Studi Lucca, Lucca, Italy
Full list of author information is available at the end of the article

Abstract

We investigate Bitcoin network observing transactions broadcasted into the network during a week from 04/05/2016 and then monitoring their inclusion into the blockchain during the following seven months.We unveil that 42% of the transactions are still not included in the Blockchain after 1 h from their appearance and 20% of the transactions are still not included in the Blockchain after 30 days, therefore revealing a great inefficiency in the Bitcoin system. However, we observe that most of these "forgotten" transactions have low values and in terms of transferred value the system is less inefficient with 93% of the transactions value being included into the Blockchain within 3 h and 98.8% within a day. The fact that a sizeable fraction of transactions is not processed timely casts serious doubts on the usability of the Bitcoin Blockchain for reliable time-stamping purposes. It also calls for a debate about the right systems of incentives which a peer-to-peer unintermediated system should introduce to promote efficient transaction recording

Keywords: Blockchain; Bitcoin; Transaction inefficiency

1 Introduction

Behind Bitcoin [1], the most popular cryptographic currency, there are users distributed all over the world who, in a voluntary way or for profit, participate in a network where transactions are announced, verified and eventually inserted into blocks of a massively replicated ledger known as Blockchain [2]. The Blockchain is a distributed database which keeps track of all transactions made by using the Bitcoin currency. Transactions involve one or more input Bitcoin addresses which are sending some funds to one or more "output" addresses. Despite the success of this new approach, which has seen Bitcoin becoming the most important cryptocurrency with capitalisation exceeding sixty billions US dollars (September 2017), the system is far from being optimal.

In the Bitcoin network the recording of transactions inside blocks and their validation is performed by special nodes called "miners". They compete to find a brute-force solution of a cryptographic task consisting in finding a hash number (associated with the block content and with the previous block hash) which is smaller than a given target number. This is the so-called "proof of work", a mechanism through which the system reaches a community consensus over the order and validity of transactions based on majority of computational power [1]. The proof of work is one of the most important elements introduced by Bitcoin, it indeed solves several issues related to trust and machine synchronisation that

are otherwise hard to manage in a distributed system operating between untrustful peers. [1–5]. Miners are incentivised by a reward consisting in a Bitcoin credit (newly issued) given to the first who finds and successfully broadcasts the valid hash.

In this paper, we measure how efficiently transactions are recorded into the blockchain by observing the Bitcoin network during the interval between May and December 2016. In that period, there were around 6000 peers participating to the Bitcoin network. Blocks contained typically between 1 and 1.7 thousand transactions, counting for about 100–170 transactions per minute. These transactions mobilised a capital of about 152 Bitcoins per minute (about 91,787 USD/minute at September 2016 exchange rate). During the period from 04/05/2016 to 11/05/2016, we have monitored the transactions announced in the network by setting up a customised client that recursively established connections with all reachable nodes. We then analysed the successive blocks in the Blockchain until to December 2016 and we measured the interval of time needed to have the transactions correctly recorded into valid blocks (thus becoming parts of the Blockchain).

As a result, we find that most of the transactions have been recorded in the Blockchain just after a few blocks. In particular 58% of transaction have been recorded within one hour. However we also observe that a rather large amount of transactions are left out for a noticeably long time interval. Specifically 20% of transactions have not been inserted into the blockchain after 30 days from their appearance. We argue that the blockchain procedure, even if very effective, it is also prone to intrinsic inefficiencies. In particular, while miners are incentivised to verify transactions by both remuneration for the proof of work and transaction fees, neither they, nor other mechanisms ensure that all transactions are actually recorded.

1.1 Blockchain

In the Bitcoin network, transactions are made and immediately announced by broadcasting them to the neighbouring network nodes that propagate the announce further. Nodes also validate transactions which are gathered into blocks which are cryptographically sealed and inserted (every 10 minutes approximately) into the Blockchain after a validation from the community. Bitcoin network participants reach consensus on the order of the transactions by voting by computation power majority (the proof of work). In theory every network node could participate to this consensus mechanism, but, in the years, this activity has become typical of a specialised part of the community called “miners”. Miners get newly emitted Bitcoin in reward for this activity. A block contains the hash of the last valid block and the record of the most recent transactions observed by the miner and not included yet in the Blockchain. The miner will try to seal it cryptographically with a hash produced from the block itself and a random part. If the hash number is by chances smaller than a threshold imposed by the proof-of-work then it is considered “valid” and it can start to be broadcasted to the network. When a node receives a new block, it should verify if the block is valid. In order to do that, it checks whether the hash of the block fulfils the proof-of-work requirements. After that, the node also verifies the digital signatures and the formatting of each transaction inside the block. If the whole block and all the transactions are verified, it accepts the new block as valid and starts propagating it through the network (and if the node is a miner, also it will start to discover the next block on top of it). Conversely, if the block is not valid, or at least one transaction inside the block is invalid, the block will be discarded. The Blockchain is the chain of blocks built one on

top of the other in chronological sequence uniquely associated with a sequence of hash numbers. Miners get their gain mainly from the cryptographic sealing of new blocks with a valid hash number; therefore, they have little incentives to make the system efficient by carefully checking if all transactions are included in the blocks.

1.2 Bitcoin communication protocol

All Bitcoin clients are connected to each other in a peer to peer network. This means that there are no central servers or authorities. Each node individually decides how to contribute to the network by choosing which service to provide. For example, by relaying transactions, by storing a copy of the Blockchain or by using their own computational power for mining. A node wanting to join to the network for the first time needs to connect to some special peers called "seeds". Such seeds provide their list of peers. This list does not depend on geographic location of clients; all the clients included are chosen randomly and the list can contain up to one thousand nodes. After retrieving the peers' lists, a node chooses amongst them until it reaches its default max number of connections (usually from 8 to 126 established connections, but the number of connections may vary according to the configuration of the Bitcoin client used and according to the network setting of the client itself). Each node, once connected to the network, can send and receive messages (such as blocks, transactions and new peers joined on the network) from all the other connected nodes. All these messages have to follow the rules (that may have different customisation) settled up by the Bitcoin Protocol [6], which consists of a set of messages used by clients to enable communication among peers.

2 Related work

In the last few years there has been some interest in the study of transaction and block propagation dynamics in the Bitcoin network with two notable contributions from Decker [7] and Miller [8]. There are also online services, such as Blockchain.info [9] which allows users to explore blocks and transaction s. Furthermore, the platform Bitnodes [10] provides snapshots of all reachable peers in the network and some statistics related to the type of the client (i.e. protocol version used, last block stored, and ip-geolocalisation). Since all the data are provided as a list of online clients, it is impossible to reconstruct how the peers are connected to each other or how information propagates among them. The approach used to discover peers in the Bitcoin network is to send recursively "getaddr" message to each reachable node in order to get back part of their known nodes list. In 2015 it has been published a Bitcoin network investigation called Coinscope [8] which used this approach in order to discover clients. This is done by introducing an algorithm, named "AddressProbe" which is able to track how peers are connected. At that time, before the release of Bitcoin Core 0.10.1 [11], it was still possible to discover the connections because each client kept updated the timestamp of a peer in a "mempool" after every exchange of data. Every time a client replied back to a peers list, it also sent their updated timestamps. The mechanism for updating the timestamp was the following: if a node exchanged some messages with a peer, it kept its own timestamp on the updated database. If instead a node discovered some new nodes through another peer, it applied a two hours penalties on the timestamp before storing the address into its own peer database. Through this mechanism it was possible to guess the connections of a peer just retrieving several time the known peers list and sorting all the records in chronological order [12]. However, it has been shown [11, 12] that reconstructing peers network could be used to attack

Bitcoin Core clients. To avoid the possibility of such attacks, the software was modified in order to avoid the updating of the timestamp of a connected client when they send or receive data. After last update on the client we noticed that, for an active connection, the timestamp is updated only when the connection drops or after 24 hours (in the case the connection is still alive). All the other cases are still as described in [8]. Finally, data propagation rate in the Bitcoin network was studied by Decker et al. in [7] where, by establishing connections with each node, they measured the time that blocks or transactions take to propagate into the network.

In this paper, we follow this methodology to identify the appearance of blocks and transactions in the network and we measure the propagation dynamics in the network and the time they take to be included inside the Blockchain.

3 Methods

To monitor the Bitcoin network we set up a customised client able to recursively establish a connection with every reachable node, and the ones in the peers list. To accomplish this goal, our client implemented only a reduced set of messages of the whole protocol:

- *getaddr, addr*—The "getaddr" messages is used to request a list of known peers from a node. The node issues an "addr" message as response. This message can contain up to one thousand known nodes. "addr" messages are also sent automatically to each connected node when the client establish a connection with a new node.
- *inv*—The "Inventory" message is sent by a client when it discovers new blocks or transactions in order to spread them on the network. In the same *"inv"* message it is possible to have both blocks and transactions.

We used *"addr"* messages in order to connect to all reachable peers (as well as to the new discovered ones once they join the network). Once connected, we stored all the inventory messages received in the form: *timestamp, address, hashcode*. Here *timestamp* is a 64 bit integer representing the time and date when the *"inv"* message was received. *address* indicates the ip address of the nodes (which can belong to ipv4, ipv6 or tor networks) and *hashcode* is the hashing string corresponding to a block or to a transaction. We established one connection to each peer without making any *"getdata"* request in order to limit the load on the network. This approach has the drawback that each peer can close the connection at any time without sending any alert. When this happened, we immediately tried to establish a new connection (the information shared with the other peers when connections are down is lost).

Data exchanged by peers consists of coordinating signals (i.e. announcing new blocks or transactions) and data messages (blocks, addresses and transactions). Data was collected by joining the network as a normal node and trying to establish a connection within each peer address discovered and waiting for *"inv"* messages for both, blocks and transactions. The client for collecting the data was coded in Go programming language [13].

4 Data

We monitored the Bitcoin network activity during the period from Wed, 04 May 2016 01:20:45 GMT to Wed, 11 May 2016 18:44:58 GMT. During this period, we observed over twelve thousands unique peers (12,424) of which 8969 belonging to ipv4 network, 3332 belonging to ipv6 network and 124 belonging to Tor network, with a number client connected at the same time ranging between 5 to 7000. This amount of peers is consistent

with the amount reported by Bitnodes [10]. Surprisingly, we received from the peers more than 126 thousands different blocks some of them valid but "old", where the oldest of them were included into the Blockchain more than 3 years earlier. Instead, the number of blocks mined during the listening period is of 1209 valid blocks (from block height 410,119 to 411,327). Overall we collected 592 GB of data with the most part regarding transactions *"inv"* messages (589 GB) while the remaining related to blocks *"inv"* messages. As regards the 12,424 different nodes that have been observed, we received blocks and transactions together only from 11,537 nodes, we received blocks from all of them, while the number of nodes with transactions information is larger and accounts to 12,168 nodes. We classified blocks and transactions as follows:

- *Blocks*

  Mined During Listening Block (MDLB)—This set identifies all the blocks included on the Blockchain during the listening period and propagated by the peers before the next block was discovered. There are 1209 blocks discovered by 530 source nodes and spread through 11,179 destination nodes. The maximum number of blocks discovered by a single node during the listening time is 86. These are the only blocks analysed.

  Echo Block (EB)—This set identifies all the blocks already included in the Blockchain and propagated in delay. We have 406,457 echo blocks, propagated from 6938 nodes.

  Fork Block (FB)—This set identifies all the blocks not included in the Blockchain even if they had a valid hash. There are 34 fork blocks of this kind.

  Invalid Block (IB)—This set identifies all the blocks not included in the Blockchain, but propagated by the peers despite having a hash above the proof-of-work threshold (not valid). There were 51,103 Invalid blocks transmitted by 23 nodes.

- *Transactions*

  Blockchain Transactions (BT)—They consist of valid transactions, included in the Blockchain, observed and propagated through the network before the block in which they were eventually included is discovered. We received 1,744,899 Blockchain transactions, from which 1,725,508 were included in a block during the listening time and 19,391 after the listening time. We have discarded transactions observed for the first time during the mining of the first and of the last block. Similarly, we have discarded also those received after they were included into a block. These transactions are noise probably originated by nodes that do not verify the validity of new transactions. There is a large number of such old transactions echoed in the system. They must be discarded since they have no relevance for the analysis we are performing. Also we did not analyse transactions with a set locktime (about five thousands). The final subset for our analysis is therefore given by 64,994 transactions generated by 2518 nodes.

  Echo Transaction (ET)—Valid Transaction, already included in a block but still propagated in delay. We have received 12,425 echo transactions that were not analysed.

  Invalid Transaction (IT)—Transaction not valid for some reasons. We received 62,889 Invalid transactions that were not analysed.

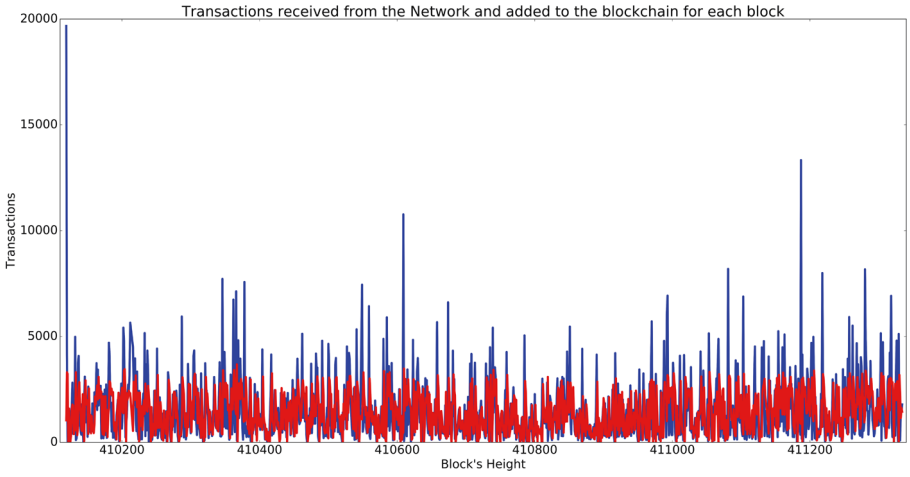


Figure 1 Transactions per block. This figure compares the number of Transactions observed during block mining (in blue) and the number of transactions included in the Blockchain during the same block mining time (in red)

5 Results and discussion

We investigated both the transaction dynamics and the block dynamics on the Bitcoin network. In Fig. 1 we compare the number of transactions observed during the listening period and the number of transactions included in the blocks during the same period. From this figure we can see that there are different dynamics. Let us now quantify the interplay between these two dynamics. Specifically we first look separately at the statistical properties of the blocks and transactions dynamics. We then look at the statistics of the process of inclusion of transaction records into valid blocks.

5.1 Block dynamics

Firstly we measured the time needed to mine the valid block hashes (MDLB). We find that the minimum time is about 2 minutes, while the maximum time is 77 minutes; the mean time for discovering a block is about 9 minutes and for the 50% percentile the time is about 6 minutes. Tables 1 and 2 report respectively the protocols and the Bitcoin client used by the nodes of the network.

When a valid block hash is discovered by a miner it is broadcasted to the whole network for validation by the other miners that consequently start the process of mining the next block on top of this one. This propagation of the block hash through the network nodes takes some time. In Fig. 2 we report, for each block, the number of nodes reached vs. the time lapse since the block was first observed in the Bitcoin network. Each trace of the plot terminates when a new valid block is discovered, for this reason the curves have different durations (around 10 min on average). We observe that different blocks reach a different number of nodes which depends on the propagation dynamics, on the number of nodes present during the propagation and on the number of active connections established by our client during the block propagation time. We observe that the typical propagation consists in a fast initial increase during the first second of propagation when about 10% of nodes are reached. In this initial phase, the average propagation law (black line in the plot) is consistent with an exponential growth for the number of nodes $N(t)$ which is what is expected for a diffusion process over a random network [14]. However, this fast propa-

Table 1 Bitcoin Protocol version used by nodes

Protocol	Number of clients
70012	6655
70002	3013
N/A	771
7000	1153
70013	88
70010	78
80001	71
70011	68
80000	24
99999	4
50400	2
70014	2
60000	1
70003	1
60002	1
80002	1

Table 2 Bitcoin Client Software used by 20 or more nodes

Bitcoin Software and version	Number of clients
Classic:0.12.0	2969
Satoshi:0.12.1	1790
Satoshi:0.12.0	1691
Satoshi:0.11.2	1323
N/A	771
Satoshi:0.11.0	368
Satoshi:0.10.2	233
Satoshi:0.11.1	226
Classic:0.11.2	184
Satoshi:0.12.99	144
Satoshi:0.11.2(bitcore)	142
Satoshi:0.9.3	122
Satoshi:0.10.0	116
BTCC:0.12.1	93
Satoshi:0.8.6	72
Satoshi:0.9.1	71
BitcoinUnlimited:0.12.0(EB16; AD4)	70
Satoshi:0.10.1	67
Satoshi:0.9.2.1	56
Satoshi:0.8.5	42
Bitcoin XT:0.11.0D	29
Bitcoin XT:0.11.0	26
Bitcoin XT:0.11.0E(Linux; x86_64)/&22',	22
Satoshi:0.8.1	20

gation is followed by a second phase that reaches about 60% of nodes within the first 10 s with a propagation that could be (too short interval for a good fitting) a power law of the kind $N(t) \propto t^{1.1}$. Then propagation it slows further down approaching saturation with the rest of the nodes reached after several minutes.

We observe that blocks first announced by some nodes propagate consistently faster (or slower) than others. We also observe that, at the extremes, blocks first propagated by the fastest node reach 50% of the peers in 2.3 s whereas blocks first propagated by the slowest node reach 50% of the peers in more than 1800 s. Client type is another factor in the propagation process; however, the limited statistics does not allow us to determine more precisely this effect.

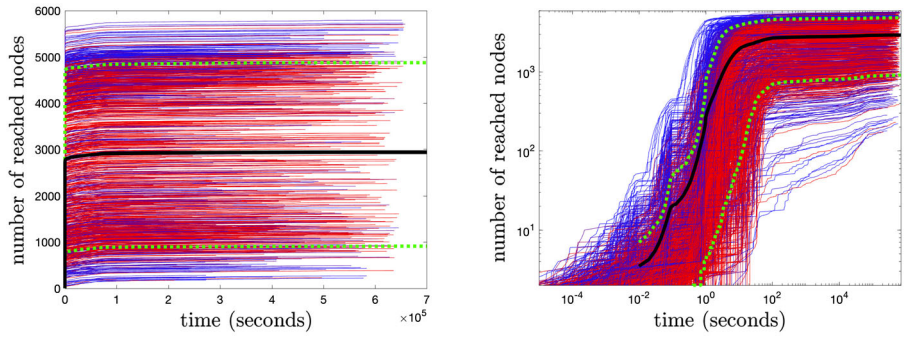


Figure 2 Blocks propagation. Number of nodes reached by a new valid block before a following block is discovered (left). The colours are associated with the size of the block: the size increases passing from blue to red. The black line is the average of all the observations and the two dashed green lines are respectively the 10% (lower) and 90% (upper) percentiles. The right plot the same of the left one but in log–log scale

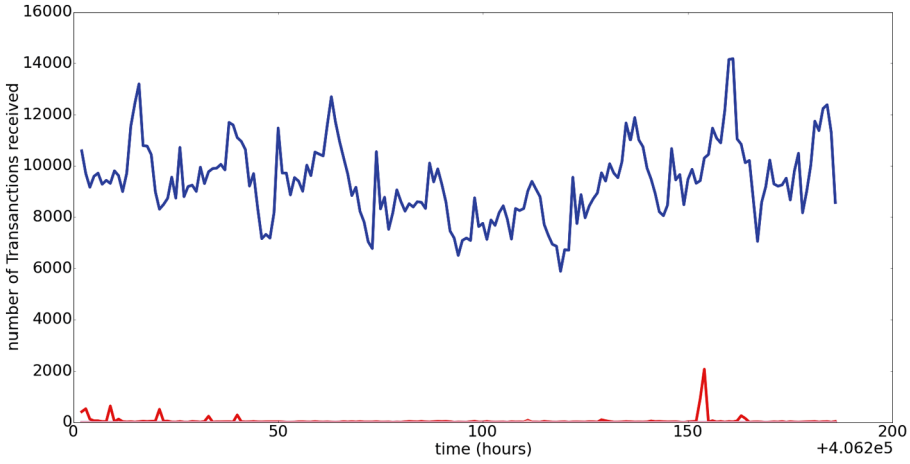


Figure 3 Unique Transaction per hour. Number of Transactions per hour received during the listening time. The blue line represent the transactions included in the Blockchain during or after the listening time (BT). The red line represent the invalid transactions (IT)

5.2 Transaction dynamics

We recorded the transaction id received from clients together with the client address, and the time. Figure 3 shows the received transactions rate per hour for IT set (in red) and BT plus ET set (in blue). We observe a fluctuating number of BT+ET transactions between 7000 and 14,000 per hour, whereas the IT have sporadic peaks that can reach 2000 but for most of the observation window do not contribute significantly to the total number of observations. During the listening time we received 1,820,212 Transactions; 1,722,696 of them were included in the Blockchain in the period until Sat, 09 Jul 2016 10:52:38 GMT. The Blockchain contains other 1266 transactions that have been produced during the monitoring period but were not observed by us in the network. An amount of 1208 of them corresponds to the zero-th transaction of each block (which is not broadcasted), whereas we could not observe the remaining 58 ones.

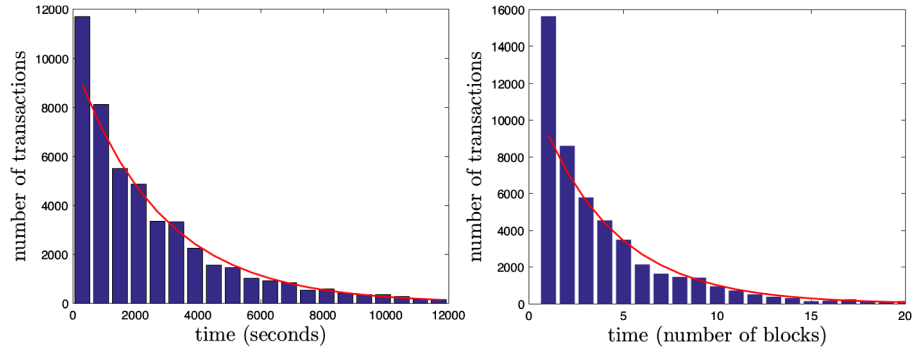
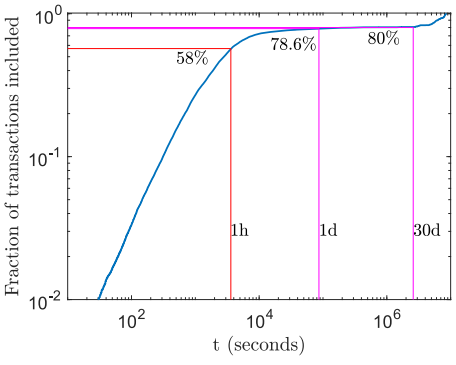


Figure 4 Transaction stand-by time. Distribution of time intervals between the first time a transaction is observed in the network and the time when it is included into a valid block. The left plot reports time in seconds and the right plot reports time in number of blocks (approx 10 min each). The red line are best fits with exponential decay law

Figure 5 Known transactions included by time. Fraction of transactions included in the Blockchain after a given amount of time (seconds, *x*-axis) from first observation in the network. The two vertical lines mark 1 h and 30 days



5.3 Inclusion of transactions into blocks

Let us now quantify which is the fraction of transactions that are recorded in the blocks mined after their first observation in the network. Specifically, we computed the interval of time between the first time a transaction is seen in the network, as received by our client, and the time when it is included in the Blockchain. Figure 4 shows the distribution of such intervals of time measured in seconds (left) and in number of blocks (right). We observe a decreasing behaviour: 24% of transactions are included in the first block mined after their first observation in the network, 13% are recorded in the successive block and 9% in the following one. The inclusion rate is consistent with a exponential decay $\sim e^{(-t/\Delta)}$, the best-fitting curve is reported in the figure with the red line. The coefficient Δ is the characteristic time and it was measured to be respectively 2800 s and 4.1 blocks. It results that for the initial decay phase (the first 20 blocks) the exponential behaviour is well followed with $R^2 = 0.98$ and $R^2 = 0.96$ respectively. However, we also observe from Fig. 4 that for longer times the empirical time distribution does no longer follow precisely an exponential decay; instead, it tends to have proportion of transactions that takes longer than expected to be included in the blockchain. These slower tails can be seen by looking at the cumulative distribution, which is reported in log–log scale in Fig. 5. We observe that 58% of the transactions are included in the Blockchain after 1 h from the first time they were seen but, remarkably, 20% of the transactions are still not included after 30 days, revealing therefore a great inefficiency in the system.

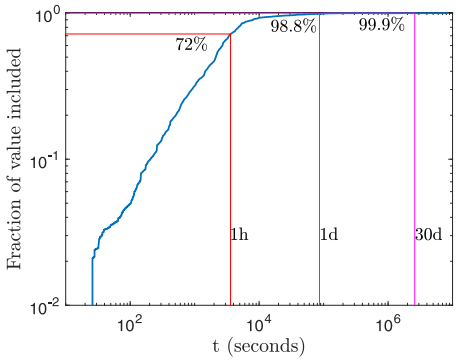


Figure 6 Average value included by time. Fraction of transferred value included in the Blockchain after a given amount of time (seconds, *x*-axis) from first observation in the network

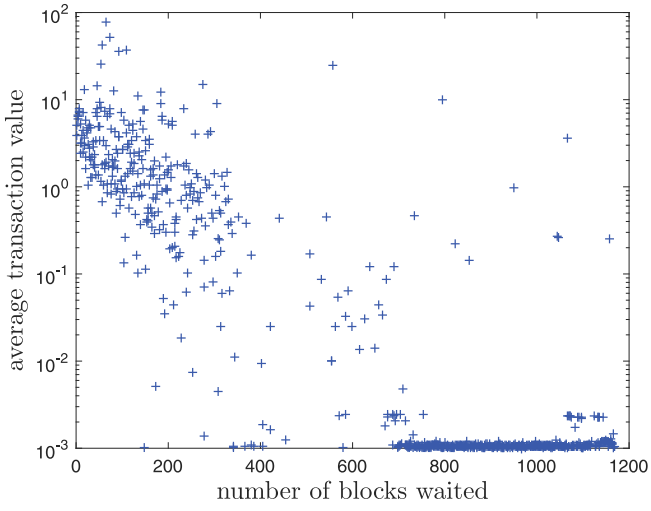


Figure 7 Value vs time. Average value of the transaction (in Bitcoins) vs. waiting time in blocks numbers

A slightly different outcome is achieved if, instead of the number of transactions, we measure the fraction of transferred value that is included in the Blockchain after a given amount of time (Fig. 6). In this case, we note that the process is still rather slow but most of the value is included in the Blockchain within 3 h (93%) and after 30 days only 0.1% of value is left to be included. This apparent inconsistency with the previous results is caused by the fact that the tail of the probability distribution for long waiting transactions is mostly populated by transactions containing only very small amounts as indicated by Fig. 7.

Fees (computed as the difference between total value inserted in the transaction minus the total value paid) play a role on the time a transaction takes to be included in the Blockchain but only for what concerns the facilitation of insertion of large transactions. We can indeed observe from Fig. 8(a) that in average larger fees-per-byte (*y* axis: fee/size) corresponds to faster inclusion times. However the spread is very large. If instead we look at the fee vs. waiting time regardless to the size of the transaction we do not observe any trend indicating that the fee facilitates speeds the transaction regardless the size. This is reported in Fig. 9 where we observe that some transactions associated with high fees have very long waiting times and, *vice-versa*, transactions with small fees are processed rather rapidly. This might indicate that the decreasing trend reported in Fig. 8(a) is mostly driven by the fact that smaller transactions tend to be included more quickly. Figure 8(b) shows

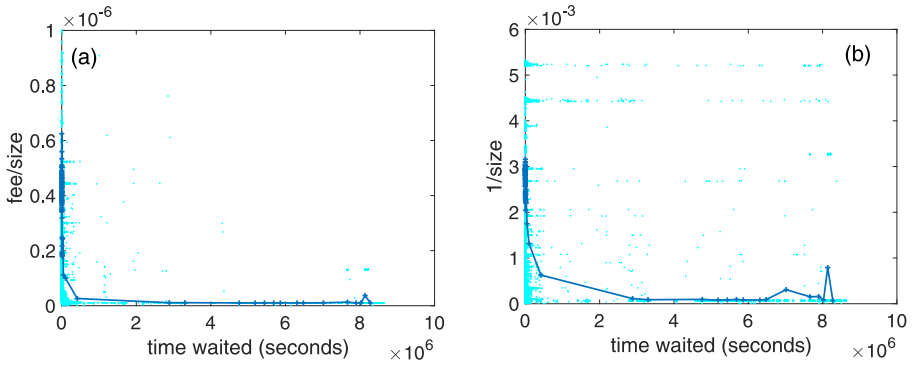


Figure 8 Fee/size and 1/size vs time. (**a**) Fee/size vs time. Fee-per-byte paid for a transaction vs. time waited before the transaction is registered in the blockchain. (**b**) 1/size vs time. Inverse of the size of a transaction vs. time waited before the transaction is registered in the blockchain

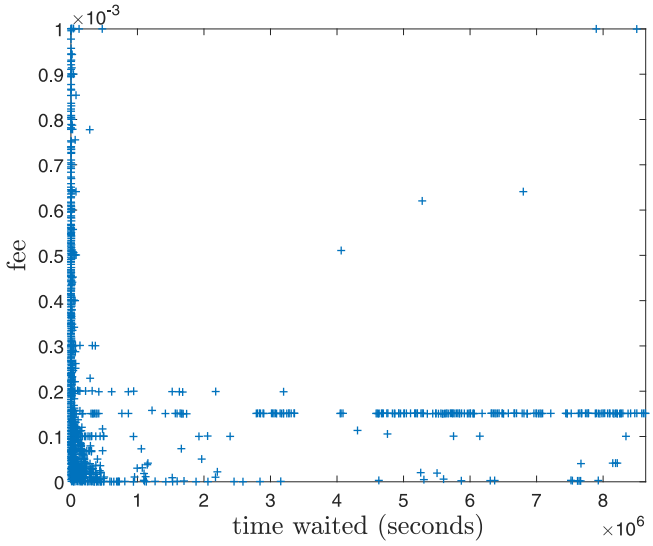


Figure 9 Fee vs time. Fee paid for a transaction vs. time waited before the transaction is registered in the blockchain

indeed that this is the case with an overall similar decreasing trend revealed in the plot of 1/size vs. waiting time.

6 Conclusions

By monitoring the Bitcoin network activity during a period of one week and by following the dynamics of inclusions of transactions within the Blockchain during the following three months we unveiled strong inefficiencies. The Bitcoin system fails in taking accurate record of the transactions with some of them taking months before being recorded in the Blockchain. This inefficiency is much larger in terms of transaction recording than in terms of volumes exchanged. Indeed, we observe that 42% transactions are were not included in the Blockchain after 1 h from their appearance and 20% of the transactions were not included in the Blockchain after 30 days. However, we observe that most of these "forgotten" transactions have low values and in terms of transferred value the system is less

inefficient with 93% of the transactions value being included into the Blockchain within 3 h. We note that this inaccurate recording did not seem to be caused by the fact that block size that at the time of observation was limited to 1 MB and only few thousands transaction can be included into a block. We indeed measured average block size 0.8 MB, with only 3% of blocks exceeding 0.99 MB band and even with some blocks without transactions [9]. As pointed out earlier, most affected transactions are those of small value, even if we observed some large transactions that have being recorded with delays of over one month. The fact that mostly small transactions are 'forgotten' keeps overall efficiency of the Bitcoin value exchange system to acceptable levels. However, this poses serious questions on the possibility of using Bitcoin as a reliable time-stamping system where small value transactions are used to register operations outside the Bitcoin system.

We conclude that such inefficiency in the Bitcoin system is most likely due to lack of sufficient incentives for peers and miners to verify, propagate and record transactions. Indeed, peer contribution to the Bitcoin network is mostly driven by mining rewards and efficient record keeping is not incentivised enough. Transaction fees should incentivize miners to process transactions timely and guarantee their recording onto the blockchain, however at the time of observation fees did not seem to play a significant role in the recording delayes.

Acknowledgements
Not applicable.

Funding
GC acknowledges support from EU projects CoeGSS (grant num. 676547), Multiplex (grant num. 317532), Openmaker (grant num. 687941), SoBigData (grant num. 654024) and the FET projects SIMPOL (grant num. 610704), DOLFINS (grant num. 640772).

Abbreviations
MDLB, Mining During Listening Block; EB, Echo Block; FB, Fork Block; IB, Invalid Block; BT, Blockchain Transaction; ET, Echo Transaction; IT, Invalid Transaction.

Availability of data and materials
Data are available for free download from the UCL Centre for Blockchain Technologies website: http://blockchain.cs.ucl.ac.uk/.

Ethics approval and consent to participate
We did not realise experiments with humans and this part is not applicable.

Competing interests
The authors declare that they have no competing interests.

Consent for publication
As from the above declaration we do not need consensus from third parts for publication.

Authors' contributions
All authors contributed equally to this work. All authors read and approved the final manuscript.

Author details
[1]IMT Alti Studi Lucca, Lucca, Italy. [2]Department of Computer Science, UCL, London, UK. [3]UCL Centre for Blockchain Technologies, London, UK. [4]Department of Mathematics, King's College, London, UK. [5]Institute of Complex Systems CNR, Dep. of Physics, University "Sapienza", Rome, Italy. [6]London Institute for Mathematical Sciences, London, UK. [7]Catchy s.r.l., Rome, Italy.

Publisher's Note

Received: 14 June 2017 Accepted: 23 August 2018 Published online: 05 September 2018

References

1. Nakamoto S (2009) Bitcoin: a peer-to-peer electronic cash system. http://www.bitcoin.org/bitcoin.pdf
2. Antonopoulos AM (2014) Mastering Bitcoin: unlocking digital crypto-currencies, 1st edn. O'Reilly Media, Inc.

3. Weber I, Xu X, Riveret R, Governatori G, Ponomarev A, Mendling J (2016) Untrusted business process monitoring and execution using blockchain. In: Business process management. Springer, Cham, pp 329–347. https://doi.org/10.1007/978-3-319-45348-4_19
4. Innovation in payment systems. https://bitcoin.org/en/innovation
5. Aste T, Tasca P, Matteo TD (2017) Future impact of blockchain technologies on services, businesses and regulation. IEEE Comput
6. Bitcoin Protocol documentation. https://en.bitcoin.it/wiki/Protocol_documentation
7. Decker C, Wattenhofer R (2013) Information propagation in the Bitcoin network. In: 13th IEEE international conference on peer-to-peer computing (P2P), Trento, Italy
8. Andrew M, James L, Andrew P, Neal G, Dave L, Neil S, Bobby B Discovering Bitcoin's public topology and influential nodes
9. Bitcoin block explorer—blockchain.info. https://blockchain.info
10. Bitnodes is currently being developed to estimate the size of the Bitcoin network by finding all the reachable nodes in the network. https://bitnodes.21.co/
11. Guessing Bitcoin's p2p connections. http://jonasnick.github.io/blog/2015/03/06/guessing-bitcoins-p2p-connections/
12. Biryukov A, Khovratovich D, Pustogarov I (2014) Deanonymisation of clients in Bitcoin P2P network. CoRR. http://arxiv.org/abs/1405.7418
13. The go programming languages. https://golang.org/
14. Caccioli GLF, Aste T (2016) Scalability and egalitarianism in peer-to-peer networks. In: Aste T, Tasca P, Pelizzon L, Penroy N (eds) Banking beyond banks and money. New economic windows. Springer, Berlin